

Trabalho 0 - Processamento de Imagens

Décio Gonçalves de Aguiar Neto¹

¹Universidade Estadual de Campinas - UNICAMP
Instituto de Computação

1. IMPLEMENTAÇÃO

A implementação apresentada neste trabalho foi feita na linguagem python em sua versão 3.6.3, e foram utilizadas as bibliotecas scypy, matplotlib e numpy para a manipulação e visualização dos dados.

Para realizar o carregamento da imagem dada como entrada foi utilizado o método `imread()` que recebe o path da imagem que será carregada e retorna uma matriz contendo os níveis de cinza de cada píxel. Por se tratar de um numpy array o mesmo já oferece métodos que retornam os valores de mínimo, máximo e a média dos valores contidos nesta estrutura assim como as dimensões da matriz, foram esses os métodos utilizados para exibir esses valores da imagem carregada no console.

Após carregada a imagem é gerado o histograma da mesma, para isso foi utilizado o método `hist()` da biblioteca `matplotlib.pyplot` que recebe como entrada a matriz de pixels e gera o vetor de intensidades da matriz que é exibido através do método `show()` da mesma biblioteca.

Para gerar o negativo da imagem é necessário inverter os valores dos pixels, para isso a operação a ser executada deveria ser:

$$M[i][j] = 255 - M[i][j] \quad (1)$$

onde M é a matriz de pixels.

Dessa forma seria necessário a implementação de dois laços para percorrer a matriz e fazer a alteração da mesma, sendo essa uma operação que teria alto custo. Contudo para otimizar essas operações a linguagem python possui o conceito de vetorização, onde podemos fazer operações entre vetores e escalares de forma mais eficiente e intuitiva, assim o código na forma vetorizada da operação 1 é escrito da seguinte forma:

$$negativo = 255 - M \quad (2)$$

A ultima operação a ser realizada sobre a imagem de entrada é uma transformação linear onde o domínio é o intervalo inteiro de $[0, 255]$ e a imagem é o intervalo de $[120, 180]$. Para que possamos aplicar essa transformação a imagem original é necessário saber os coeficientes angular e linear da função de transformação, que são obtidos através de um sistema de equações lineares fazendo uso do domínio e imagem fornecidos, para que possam ser aplicados na matriz original e resultar na matriz transformada. A Equação 3 mostra a operação realizada sobre a matriz de pixels da imagem original onde M é a matriz da imagem original e M_t é a matriz da imagem transformada.

$$M_t = \frac{60}{255} * M + 120 \quad (3)$$

Da mesma forma que está escrito na equação, podemos utilizar o conceito de vetorização do python para escrever a transformação de mesma forma em nosso código e assim obter a imagem transformada.

Vale destacar que para que a imagem transformada seja visualizada de forma correta foi necessário setar dois parâmetros, o $vmin = 0$ e o $vmax = 255$ do método `imshow()` assim sendo possível de forma visual ver os efeitos da transformação sobre a imagem. Caso os valores não sejam setados o método `imshow()` exibe a imagem mas ajusta seus limites de exibição, assim impossibilitando de perceber a mudança visual na imagem.

2. EXECUÇÃO

Para que o código seja executado de forma correta basta executar o seguinte comando:

```
python3 T0.py caminho_da_imagem
```

A saída será dada da seguinte forma:

1. Será exibido no console informações da imagem:
 - (a) largura
 - (b) altura
 - (c) intensidade mínima
 - (d) intensidade máxima
 - (e) intensidade média
2. Será exibido em uma janela:
 - (a) imagem original
 - (b) negativo da imagem
 - (c) imagem transformada
 - (d) histograma da imagem original

Figura 1. SAÍDA DO ALGORITMO

